1 Typing Rules

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Typing rules for expressions

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t} \quad \text{TY\_VAR\_LIN}$$

$$\frac{x : t \in \Delta}{\Theta; \Delta; \cdot \vdash x : t} \quad \text{TY\_VAR}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma', x : t \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \textbf{let}\, x = e\, \textbf{in}\, e' : t'} \quad \text{TY\_LET}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : \textbf{unit}} \quad \text{TY\_UNIT\_INTRO}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : \textbf{unit} \\ \Theta; \Delta; \Gamma' \vdash e' : t\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \textbf{let}\, () = e\, \textbf{in}\, e' : t} \quad \text{TY\_UNIT\_ELIM}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \textbf{true} : \textbf{bool}} \quad \text{TY\_BOOL\_TRUE}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \textbf{false} : \textbf{bool}} \quad \text{TY\_BOOL\_FALSE}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : !\textbf{bool} \\ \Theta; \Delta; \Gamma' \vdash e_1 : t' \\ \Theta; \Delta; \Gamma' \vdash e_2 : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \textbf{if}\, e\, \textbf{then}\, e_1\, \textbf{else}\, e_2 : t} \quad \text{TY\_BOOL\_ELIM}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash k : \textbf{int}} \quad \text{TY\_INT\_INTRO}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash el : \textbf{elt}} \quad \text{TY\_ELT\_INTRO}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \cdot \vdash v : t \\ v \neq l\end{array}}{\Theta; \Delta; \cdot \vdash \textbf{Many}\, v : !t} \quad \text{TY\_BANG\_INTRO}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : !t \\ \Theta; \Delta, x : t; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \textbf{let}\, \textbf{Many}\, x = e\, \textbf{in}\, e' : t'} \quad \text{TY\_BANG\_ELIM}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash (e, e') : t \otimes t'} \quad \text{TY\_PAIR\_INTRO}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e_{12} : t_1 \otimes t_2 \\ \Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e : t\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \textbf{let}\, (a, b) = e_{12}\, \textbf{in}\, e : t} \quad \text{TY\_PAIR\_ELIM}$$

$$\frac{\begin{array}{c}\Theta \vdash t'\ \mathsf{Type} \\ \Theta; \Delta; \Gamma, x : t' \vdash e : t\end{array}}{\Theta; \Delta; \Gamma \vdash \textbf{fun}\, x : t' \rightarrow e : t' \multimap t} \quad \text{TY\_LAMBDA}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : t' \multimap t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash e\, e' : t} \quad \text{TY\_APP}$$

$$\frac{\Theta, fc; \Delta; \Gamma \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\, fc \to e : \forall fc.t} \quad \textsc{Ty\_Gen}$$

$$\frac{\begin{array}{l}\Theta \vdash f\ \mathsf{Cap} \\ \Theta; \Delta; \Gamma \vdash e : \forall fc.t\end{array}}{\Theta; \Delta; \Gamma \vdash e[f] : t[f/fc]} \quad \textsc{Ty\_Spc}$$

$$\frac{\Theta; \Delta, g : t \multimap t'; \cdot, x : t \vdash e : t'}{\Theta; \Delta; \cdot \vdash \mathbf{fix}\, (g, x : t, e : t') : !(t \multimap t')} \quad \textsc{Ty\_Fix}$$

2 Operational Semantics

$\boxed{\langle \sigma, e \rangle \to \mathit{StepsTo}}$ operational semantics

$$\frac{}{\langle \sigma, \textbf{let}\,() = ()\,\textbf{in}\,e \rangle \to \langle \sigma, e \rangle}\ \textsc{Op\_Let\_Unit}$$

$$\frac{}{\langle \sigma, \textbf{let}\,x = v\,\textbf{in}\,e \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_Let\_Var}$$

$$\frac{}{\langle \sigma, \textbf{if}\,(\textbf{Many}\,\textbf{true})\,\textbf{then}\,e_1\,\textbf{else}\,e_2 \rangle \to \langle \sigma, e_1 \rangle}\ \textsc{Op\_If\_True}$$

$$\frac{}{\langle \sigma, \textbf{if}\,(\textbf{Many}\,\textbf{false})\,\textbf{then}\,e_1\,\textbf{else}\,e_2 \rangle \to \langle \sigma, e_2 \rangle}\ \textsc{Op\_If\_False}$$

$$\frac{}{\langle \sigma, \textbf{let}\,\textbf{Many}\,x = \textbf{Many}\,v\,\textbf{in}\,e \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_Let\_Many}$$

$$\frac{}{\langle \sigma, \textbf{let}\,(a, b) = (v_1, v_2)\,\textbf{in}\,e \rangle \to \langle \sigma, e[a/v_1][b/v_2] \rangle}\ \textsc{Op\_Let\_Pair}$$

$$\frac{e_1 = e[g/\textbf{fun}\,x : t \to \textbf{let}\,\textbf{Many}\,g = \textbf{fix}\,(g, x : t, e : t')\,\textbf{in}\,g\,x]}{\langle \sigma, \textbf{let}\,\textbf{Many}\,g = \textbf{fix}\,(g, x : t, e : t')\,\textbf{in}\,e' \rangle \to \langle \sigma, e'[g/\textbf{fun}\,x : t \to e_1] \rangle}\ \textsc{Op\_Let\_Fix}$$

$$\frac{}{\langle \sigma, (\textbf{fun}\,fc \to v)[f] \rangle \to \langle \sigma, v[fc/f] \rangle}\ \textsc{Op\_Frac\_Cap}$$

$$\frac{}{\langle \sigma, (\textbf{fun}\,x : t \to e)\,v \rangle \to \langle \sigma, e[x/v] \rangle}\ \textsc{Op\_App}$$

$$\frac{\langle \sigma, e \rangle \to \langle \sigma', e' \rangle}{\langle \sigma, C[e] \rangle \to \langle \sigma, C[e'] \rangle}\ \textsc{Op\_Context}$$

$$\frac{\langle \sigma, e \rangle \to \textbf{err}}{\langle \sigma, C[e] \rangle \to \textbf{err}}\ \textsc{Op\_Context\_Err}$$

$$\frac{0 \leq k_1, k_2}{\langle \sigma, \textbf{matrix}\,k_1\,k_2 \rangle \to \langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, l \rangle}\ \textsc{Op\_Matrix}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_1 m_{k_1,k_2}\}, \textbf{free}\,l \rangle \to \langle \sigma, () \rangle}\ \textsc{Op\_Free}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}, \textbf{share}\,l \rangle \to \langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, (l, l) \rangle}\ \textsc{Op\_Share}$$

$$\frac{f \leq 1}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, \textbf{unshare}\,l\,l \rangle \to \langle \sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}, l \rangle}\ \textsc{Op\_Unshare\_Eq}$$

$$\frac{l \neq l'}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\} \uplus \{l' \mapsto_{\frac{1}{2}\cdot f} m_{k_1,k_2}\}, \textbf{unshare}\,l\,l' \rangle \to \textbf{err}}\ \textsc{Op\_Unshare\_Neq}$$

$$\frac{\sigma' = \sigma \uplus \{l_1 \mapsto_{fc_1} m1_{k_1,k_2}\} \uplus \{l_2 \mapsto_{fc_2} m2_{k_2,k_3}\}}{\langle \sigma' \uplus \{l_3 \mapsto_1 m1_{k_1,k_3}\}, \textbf{gemm}\,l_1\,l_2\,l_3 \rangle \to \langle \sigma' \uplus \{l_3 \mapsto_1 (m_1\,m_2 + m_3)_{k_1,k_3}\}, ((l_1, l_2), l_3) \rangle}\ \textsc{Op\_Gemm\_Match}$$

$$\frac{\begin{array}{l} k_2 \neq k_2' \\ \sigma' = \sigma \uplus \{l_1 \mapsto_{fc_1} m1_{k_1,k_2}\} \uplus \{l_2 \mapsto_{fc_2} m2_{k_2',k_3}\} \end{array}}{\langle \sigma' \uplus \{l_3 \mapsto_1 m1_{k_1,k_3}\}, \textbf{gemm}\,l_1\,l_2\,l_3 \rangle \to \textbf{err}}\ \textsc{Op\_Gemm\_Mismatch}$$

3 Interpretation

$$\mathcal{V}_k[\![\mathbf{unit}]\!] = \{(\emptyset, *)\}$$

$$\mathcal{V}_k[\![\mathbf{bool}]\!] = \{(\emptyset, true), (\emptyset, false)\}$$

$$\mathcal{V}_k[\![\mathbf{int}]\!] = \{(\emptyset, n) \mid 2^{-63} \leq n \leq 2^{63} - 1\}$$

$$\mathcal{V}_k[\![\mathbf{elt}]\!] = \{(\emptyset, f) \mid f \text{ a IEEE Float64 }\}$$

$$\mathcal{V}_k[\![f\,\mathbf{mat}]\!] = \{(\{l \mapsto_{2^{-f}} \_\}, l)\}$$

$$\begin{aligned}\mathcal{V}_k[\![!(t' \multimap t'')]\!] = {} & \{(\emptyset, \mathbf{Many}\, v) \mid (\emptyset, v) \in \mathcal{V}_k[\![t' \multimap t'']\!]\} \\ & \cup \{(\emptyset, \mathbf{fix}(g, x : t, e : t')) \mid \forall\, j < k, (\sigma, v) \in \mathcal{V}_j[\![t]\!]. \\ & \qquad (\sigma, \mathbf{let\,Many}\, g = \mathbf{fix}\,(g, x : t, e : t')\,\mathbf{in}\, g\, v) \in \mathcal{C}_j[\![t']\!]\}\end{aligned}$$

$$\mathcal{V}_k[\![!t]\!] = \{(\emptyset, \mathbf{Many}\, v) \mid \neg(\exists t', t''.\ t = t' \multimap t'') \wedge (\emptyset, v) \in \mathcal{V}_k[\![t]\!]\}$$

$$\mathcal{V}_k[\![\forall fc.\ t]\!] = \{(\sigma, \mathbf{fun}\, fc \to\ v) \mid \forall f.\ (\sigma, (\mathbf{fun}\, fc \to\ v)\,[f]) \in \mathcal{V}_k[\![t[fc/f]]\!]\}$$

$$\mathcal{V}_k[\![t_1 \otimes t_2]\!] = \{(\sigma_1 \star \sigma_2, (v_1, v_2)) \mid (\sigma_1, v_1) \in \mathcal{V}_k[\![t_1]\!] \wedge (\sigma_2, v_2) \in \mathcal{V}_k[\![t_2]\!]\}$$

$$\begin{aligned}\mathcal{V}_k[\![t \multimap t']\!] = {} & \{(\sigma, \mathbf{fun}\, x : t \to e) \mid \forall j < k, (\sigma_v, v) \in \mathcal{V}_j[\![t]\!].\ \sigma \star \sigma_v \text{ defined } \Rightarrow \\ & \qquad (\sigma \star \sigma_v, (\mathbf{fun}\, x : t \to e)\, v) \in \mathcal{C}_j[\![t']\!]\}\end{aligned}$$

$$\begin{aligned}\mathcal{C}_k[\![t]\!] = {} & \{(\sigma_s, e) \mid \forall\, j \leq k, \sigma_r.\ \sigma_s \star \sigma_r \text{ defined } \Rightarrow \langle \sigma_s \star \sigma_r, e\rangle \to^j \mathbf{err}\ \vee \exists \sigma_f, e'. \\ & \qquad \langle \sigma_s \star \sigma_r, e\rangle \to^j \langle \sigma_f \star \sigma_r, e'\rangle \wedge (e' \text{ is a value } \Rightarrow (\sigma_f \star \sigma_r, e') \in \mathcal{V}_{k-j}[\![t]\!])\}\end{aligned}$$

$$\mathcal{I}_k[\![\cdot]\!]\theta = \{[]\}$$

$$\mathcal{I}_k[\![\Delta, x : t]\!]\theta = \{\delta[x \mapsto v_x] \mid \delta \in \mathcal{I}_k[\![\Delta]\!]\theta \wedge (\emptyset, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]\}$$

$$\mathcal{L}_k[\![\cdot]\!]\theta = \{(\emptyset, [])\}$$

$$\mathcal{L}_k[\![\Gamma, x : t]\!]\theta = \{(\sigma \star \sigma_x, \gamma[x \mapsto v_x]) \mid (\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta \wedge (\sigma_x, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]\}$$

$$\begin{aligned}[\![\Theta; \Delta; \Gamma \vdash e : t]\!] = {} & \forall \theta, k, \delta, \gamma, \sigma.\ \mathrm{dom}(\Theta) = \mathrm{dom}(\theta) \wedge (\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta \wedge \delta \in \mathcal{I}_k[\![\Delta]\!]\theta \Rightarrow \\ & \qquad (\sigma, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t)]\!]\end{aligned}$$

4 Soundness Proof

$$\forall \Theta, \Delta, \Gamma, e, t.\ \Theta; \Delta; \Gamma \vdash e : t \Rightarrow [\![\Theta; \Delta; \Gamma \vdash e : t]\!]$$

Proof sketch: Induction over the typing judgements.

Assume: 1. Arbitrary $\Theta, \Delta, \Gamma, e, t$ such that $\Theta; \Delta; \Gamma \vdash e : t$.
2. Arbitrary $\theta, k, \delta, \gamma, \sigma$ such that:
   a. $\text{dom}(\Theta) = \text{dom}(\theta)$
   b. $(\sigma, \gamma) \in \mathcal{L}_k[\![\Gamma]\!]\theta$
   c. $\delta \in \mathcal{I}_k[\![\Delta]\!]\theta$.
3. W.l.o.g., all variables are distinct/dom(Δ) and dom(Γ) are disjoint.
4. And so that over expressions $\gamma \circ \delta = \delta \circ \gamma$.
5. By construction, $\text{dom}(\Delta) = \text{dom}(\delta)$ and $\text{dom}(\Gamma) = \text{dom}(\gamma)$.
6. ??? $\mathcal{V}_k[\![\theta(t)]\!] \subseteq \mathcal{C}_k[\![\theta(t)]\!]$.
7. ??? "Stronger heap"/frame rule: $\langle \sigma, e \rangle \rightarrow^{*} = \langle \sigma \star \sigma_r, e \rangle \rightarrow^{*}$.
8. ??? $\delta(\gamma(v))$ is a value.
9. ??? $j \leq k \Rightarrow {}_{-\,k}[\![\cdot]\!] \subseteq {}_{-\,j}[\![\cdot]\!]$
10. ??? $(\sigma', e') \in \mathcal{C}_{k-1}[\![\cdot]\!] \wedge \langle \sigma, e \rangle \rightarrow \langle \sigma', e' \rangle \Rightarrow (\sigma, e) \in \mathcal{C}_k[\![\cdot]\!]$

Prove: $(\sigma, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
Assume: Arbitrary $j \leq k$ and σ_r.
Suffices: Show whole expression either reduces to **err** or takes j steps.

$\langle 1 \rangle 1$. Case: Ty_Let.
Prove: $(\sigma, \gamma(\delta(\textbf{let}\, x\ =\ e\, \textbf{in}\, e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
Suffices: $(\sigma, \textbf{let}\, x\ =\ \gamma(\delta(e))\, \textbf{in}\, \gamma(\delta(e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.

$\langle 2 \rangle 1$. By induction,
1. $[\![\Theta; \Delta; \Gamma \vdash e : t]\!]$
2. $[\![\Theta; \Delta; \Gamma', x : t \vdash e' : t']\!]$.

$\langle 2 \rangle 2$. By 2b and induction on Γ', we know there exist $\sigma_{e'}$, $(\sigma_e, \gamma_e) \in \mathcal{L}_k[\![\Gamma]\!]$, such that $\sigma = \sigma_e \star \sigma_{e'}$.

$\langle 2 \rangle 3$. So, using them, θ, k, δ, and 3 we have $(\sigma_e, \gamma_e(\delta(e))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

$\langle 2 \rangle 4$. By 3, $(\sigma_e, \gamma(\delta(e))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

$\langle 2 \rangle 5$. By definition of $\mathcal{C}_k[\![\cdot]\!]$ and $\langle 2 \rangle 2$, we instantiate with j and $\sigma_r = \sigma_{e'}$ to conclude that $\langle \sigma, \gamma(\delta(e)) \rangle$ either reduces to **err** or another heap and expression.

$\langle 2 \rangle 6$. Case: **err**
??? By Op_Context_Err and 3, the whole expression reduces to **err** in $j \leq k$ steps. Since $j \leq k$ and σ_r (for 7) are arbitrary, $(\sigma, \gamma(\delta(\textbf{let}\, x\ =\ e\, \textbf{in}\, e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.

$\langle 2 \rangle 7$. Case: j steps to another heap and expression.
By Op_Context and 3, the whole expression does the same.

$\langle 2 \rangle 8$. If it is not a value, we are done. ??? If it is $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(t)]\!]$ by 8.
Suffices: $(\sigma_{ef} \star \sigma_{e'}, \textbf{let}\, x\ =\ v\, \textbf{in}\, \gamma(\delta(e'))) \in \mathcal{C}_{k-j}[\![\theta(t')]\!]$.
Suffices: ??? $(\sigma_{ef} \star \sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

$\langle 2 \rangle 9$. Define: $\gamma_{e'}(y) = v$ if $y = x$ and $\gamma(y)$ if $y \in \text{dom}(\Gamma')$.

??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}) \in \mathcal{L}_k[\![\Gamma', x : t]\!]\theta \subseteq \mathcal{L}_{k-j-1}[\![\Gamma', x : t]\!]\theta$.

$\langle 2\rangle 10$. Instantiate 2 of step $\langle 2\rangle 1$ with $\theta, k-j-1, \delta, \gamma_{e'}, \sigma_{e'}$ to conclude $(\sigma_{e'}, \gamma_{e'}(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

$\langle 2\rangle 11$. By 3, we have $\gamma(\delta(e'))[x/v] = \gamma_{e'}(\delta(e'))$ and by 7 we conclude $(\sigma_{ef} \star \sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$

$\langle 1\rangle 2$. Case: Ty_Pair_Elim.
Prove: $(\sigma, \gamma(\delta(\mathbf{let}\,(a, b) \ = \ e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t')]\!]$.
Proof: Similar to Ty_Let but with Op_Let_Pair

$\langle 2\rangle 1$. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(t_1) \otimes \theta(t_2)]\!]$, we have $v = (v_1, v_2)$.

$\langle 2\rangle 2$. Suffices: ??? $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

$\langle 2\rangle 3$. Define: $\gamma_{e'}$ to be the restriction of γ to $\mathrm{dom}(\Gamma')$.
??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}[a \mapsto v_1, b \mapsto v_2]) \in \mathcal{L}_k[\![\Gamma', a : t_1, b : t_2]\!]\theta$
$\subseteq \mathcal{L}_{k-j-1}[\![\Gamma', a : t_1, b : t_2]\!]\theta$.

$\langle 2\rangle 4$. Instantiate $[\![\Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e' : t']\!]$ with $\theta, k-j-1, \delta, \gamma_{e'}[a \mapsto v_1, b \mapsto v_2], \sigma_{e'}$.

$\langle 2\rangle 5$. ??? By 3 $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

$\langle 1\rangle 3$. Case: Ty_Bang_Elim.
Prove: $(\sigma, \gamma(\delta(\mathbf{let\,Many}\,x \ = \ e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
Proof sketch: Similar to Ty_Let, but with the following key differences.

$\langle 2\rangle 1$. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\theta(!t)]\!]$, since $\mathcal{V}_{k-j}[\![\theta(!t)]\!] = \mathcal{V}_{k-j}[\![!\theta(t)]\!]$, we have $\sigma_{ef} = \emptyset$ and $v = \mathbf{Many}\,v'$ for some $(\emptyset, v') \in \mathcal{V}_{k-j}[\![\theta(t)]\!]$.

$\langle 2\rangle 2$. Suffices: $(\sigma_{e'}, \mathbf{let\,Many}\,x \ = \ \mathbf{Many}\,v'\,\mathbf{in}\,\gamma(\delta(e'))) \in \mathcal{C}_{k-j}[\![\theta(t)]\!]$.

$\langle 2\rangle 3$. Suffices: $(\sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 2\rangle 4$. Define: $\gamma_{e'}$ as the restriction of γ to $\mathrm{dom}(\Gamma')$.

$\langle 2\rangle 5$. Instantiate $[\![\Theta; \Delta, x : t, \Gamma' \vdash e' : t']\!]$ with $\theta, k-j-1, \delta_{e'} = \delta[x \mapsto v'], \gamma_{e'}, \sigma_{e'}$ to conclude $(\sigma_{e'}, \gamma_{e'}(\delta_{e'}(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 2\rangle 6$. ??? By 3, $(\sigma_{e'}, \gamma(\delta(e'))[x/v]) \in \mathcal{C}_{k-j-1}[\![\theta(t)]\!]$.

$\langle 1\rangle 4$. Case: Ty_Unit_Elim.
Prove: $(\sigma, \gamma(\delta(\mathbf{let}\,() \ = \ e\,\mathbf{in}\,e'))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
Proof: Similar to Ty_Let but with Op_Let_Unit.

$\langle 2\rangle 1$. When $(\sigma_{ef}, v) \in \mathcal{V}_{k-j}[\![\mathbf{unit}]\!]$, we have $\sigma_{ef} = \emptyset$ and $v = ()$.

$\langle 2\rangle 2$. Suffices: ??? $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$ by 10.

$\langle 2\rangle 3$. Define: $\gamma_{e'}$ to be the restriction of γ to $\mathrm{dom}(\Gamma')$.
??? Thus, by 9, $(\sigma_{e'}, \gamma_{e'}) \in \mathcal{L}_k[\![\Gamma']\!]\theta \subseteq \mathcal{L}_{k-j-1}[\![\Gamma']\!]\theta$.

$\langle 2\rangle 4$. Instantiate $[\![\Theta; \Delta; \Gamma' \vdash e' : t']\!]$ with $\theta, k-j-1, \delta, \gamma_{e'}, \sigma_{e'}$.

$\langle 2\rangle 5$. ??? By 3 $(\sigma_{e'}, \gamma(\delta(e'))) \in \mathcal{C}_{k-j-1}[\![\theta(t')]\!]$.

⟨1⟩5. CASE: TY_BOOL_ELIM.
PROVE: $(\sigma, \gamma(\delta(\textbf{if}\, e\, \textbf{then}\, e_1\, \textbf{else}\, e_2))) \in \mathcal{C}_k[\![\theta(t)]\!]$.
PROOF: Similar to TY_UNIT_ELIM but with OP_IF_{TRUE,FALSE} and $\sigma_{ef} = \emptyset$ and $v = \textbf{Many true}$ or $v = \textbf{Many false}$.

⟨1⟩6. CASE: TY_BANG_INTRO.
PROVE: $(\sigma, \gamma(\delta(\textbf{Many}\, e))) \in \mathcal{C}_k[\![\theta(!t)]\!]$.
SUFFICES: $(\sigma, \textbf{Many}\, \gamma(\delta(e))) \in \mathcal{C}_k[\![!\theta(t)]\!]$.

⟨2⟩1. By assumption of TY_BANG_INTRO, $e = v$ for some value $v \neq l$, $\Gamma = \emptyset$ and so $[\![\Theta; \Delta; \cdot \vdash v : t]\!]$ by induction.

⟨2⟩2. SUFFICES: $(\emptyset, \textbf{Many}\, \delta(v)) \in \mathcal{C}_k[\![!\theta(t)]\!]$ by 3 and 2b.

⟨2⟩3. Instantiate $[\![\Theta; \Delta; \cdot \vdash v : t]\!]$ with $\theta, k, \delta, \gamma = [], \sigma = \emptyset$ to obtain $(\emptyset, \delta(v)) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨2⟩4. Instantiate $(\emptyset, \delta(v)) \in \mathcal{C}_k[\![\theta(t)]\!]$ with $j = 0$, and $\sigma_r = \emptyset$, to conclude $(\emptyset, v) \in \mathcal{V}_k[\![\theta(t)]\!]$.

⟨2⟩5. ??? By definition of $\mathcal{V}_k[\![!\theta(t)]\!]$, 8 and 6 we have $(\emptyset, \textbf{Many}\, \delta(v)) \in \mathcal{C}_k[\![!\theta(t)]\!]$.

⟨1⟩7. CASE: TY_PAIR_INTRO.
PROVE: $(\sigma, \gamma(\delta(\,(e, e')\,))) \in \mathcal{C}_k[\![\theta(t \otimes t')]\!]$.
ASSUME: Arbitrary $j \leq k$ and σ_r.
SUFFICES: Show whole expression either reduces to **err** or a heap and expression in j steps.

⟨2⟩1. DEFINE: $(\sigma_1, \gamma_1) \in \mathcal{L}_j[\![\Gamma]\!]$ similar to (σ_e, γ_e) in TY_LET.

⟨2⟩2. By induction,
1. $[\![\Theta; \Delta; \Gamma_1 \vdash e_1 : t_1]\!]$
2. $[\![\Theta; \Delta; \Gamma_2 \vdash e_2 : t_2]\!]$.

⟨2⟩3. Instantiate the first with $\theta, j, \delta, \gamma_1, \sigma_1$.

⟨2⟩4. Therefore, $(\sigma_1, \gamma_1(\delta(e_1))) \in \mathcal{C}_j[\![\theta(t)]\!]$.

⟨2⟩5. So, $(\sigma_1 \star \sigma_2, \gamma_1(\delta(e_1)))$ either reduces to **err** or a heap and expression in j steps.

⟨2⟩6. CASE: **err**
??? By OP_CONTEXT_ERR and 3, so too does the whole expression. Since $j \leq k$ and σ_r (for 7) are arbitrary, $(\sigma, \gamma(\delta(\,(e, e')\,))) \in \mathcal{C}_k[\![\theta(t \otimes t')]\!]$.

⟨2⟩7. CASE: j steps to another heap and expression.
By OP_CONTEXT and 3, the whole expression does the same.

⟨2⟩8. If it is not a value, we are done. ??? If it is $(\sigma_{1f}, v_1) \in \mathcal{V}_{k-j}[\![\theta(t_1)]\!]$ by 8.
SUFFICES: ??? By 10, $(\sigma_{1f} \star \sigma_{e_2},\, (v_1, e_2)\,) \in \mathcal{C}_{k-j}[\![\theta(t_1 \otimes t_2)]\!]$.

⟨2⟩9. Instantiate the second IH with $\theta, j, \delta, \gamma_2, \sigma_2$ defined as per usual.

⟨2⟩10. So, $(\sigma_{1f} \star \sigma_2, \gamma_2(\delta(e_2)))$ either reduces to **err** or a heap and expression in j steps.

⟨2⟩11. CASE: **err**
??? By OP_CONTEXT_ERR, 3, so too does the whole expression. Since $j \leq k$ and σ_r (for 7) are arbitrary, $(\sigma_{1f} \star \sigma_{e_2},\, (v_1, e_2)\,) \in \mathcal{C}_{k-j}[\![\theta(t_1 \otimes t_2)]\!]$.

⟨2⟩12. CASE: j steps to another heap and expression.

By OP_CONTEXT and 3, the whole expression does the same.

⟨2⟩13. If it is not a value, we are done. ??? If it is $(\sigma_{2f}, v_2) \in \mathcal{V}_{k-j}[\![\theta(t_2)]\!]$ by 8.
SUFFICES: ??? By 10, $(\sigma_{1f} \star \sigma_{2f},\, (v_1, v_2)\,) \in \mathcal{C}_{k-2j}[\![\theta(t_1 \otimes t_2)]\!]$.

⟨2⟩14. ??? By 9 and 6, $(\sigma_{1f} \star \sigma_{2f},\, (v_1, v_2)\,) \in \mathcal{V}_{k-j}[\![\cdot]\!] \subseteq \mathcal{V}_{k-2j}[\![\cdot]\!] \subseteq \mathcal{C}_{k-2j}[\![\cdot]\!]$ as needed.

⟨1⟩8. CASE: TY_LAMBDA.
PROVE: $(\sigma, \gamma(\delta(\mathbf{fun}\, x : t \to e))) \in \mathcal{C}_k[\![\theta(t \multimap t')]\!]$.
SUFFICES: ??? By 6, to show $\ldots \in \mathcal{V}_k[\![\theta(t \multimap t')]\!]$.
ASSUME: Arbitrary $j < k$, $(\sigma_v, v) \in \mathcal{V}_j[\![\theta(t)]\!]$ such that $\sigma \star \sigma_v$ is defined.
SUFFICES: $(\sigma \star \sigma_v, \gamma(\delta(\mathbf{fun}\, x : t \to e))v) \in \mathcal{C}_j[\![\theta(t')]\!]$.
SUFFICES: $(\sigma \star \sigma_v, \gamma(\delta(e))[x/v]) \in \mathcal{C}_j[\![\theta(t')]\!]$.

⟨2⟩1. By induction, $[\![\Theta; \Delta; \Gamma, x : t \vdash e]\!]$.

⟨2⟩2. Instantiate it $\theta, j-1, \gamma[x \mapsto v], \sigma_v \star \sigma$.

⟨2⟩3. Hence, $(\sigma_v \star \sigma, \gamma[x \mapsto v](\delta(e))) \in \mathcal{C}_{j-1}[\![\theta(t)]\!]$.

⟨2⟩4. ??? By 3, we are done.

⟨1⟩9. CASE: TY_APP.
PROVE: $(\sigma, \gamma(\delta(\, e\, e'\,))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

⟨1⟩10. CASE: TY_GEN.
PROVE: $(\sigma, \gamma(\delta(\mathbf{fun}\, fc \to e))) \in \mathcal{C}_k[\![\theta(\forall\, fc.\, t)]\!]$.

⟨1⟩11. CASE: TY_SPC.
PROVE: $(\sigma, \gamma(\delta(e\,[f]))) \in \mathcal{C}_k[\![\theta(t\,[fc/f])]\!]$.

⟨1⟩12. CASE: TY_FIX.
PROVE: $(\sigma, \gamma(\delta(\mathbf{fix}(g, x : t, e : t')))) \in \mathcal{C}_k[\![\theta(!(t \multimap t'))]\!]$.
SUFFICES: ??? to show $\ldots \in \mathcal{V}_k[\![!(\theta(t) \multimap \theta(t'))]\!]$, by 6.

⟨2⟩1. ASSUME: Arbitrary $j < k$ and $(\sigma, v) \in \mathcal{V}_j[\![\theta(t)]\!]$.

⟨2⟩2. SUFFICES: $(\sigma, \mathbf{let\, Many}\; g = \mathbf{fix}\,(g, x : t, e : t')\, \mathbf{in}\, g\, v) \in \mathcal{C}_j[\![\theta(t')]\!]$.

⟨2⟩3. LET: $e_1 = e[g/\mathbf{fun}\, x : t \to \mathbf{let\, Many}\; g = \mathbf{fix}\,(g, x : t, e : t')\, \mathbf{in}\, g\, x]$.

⟨2⟩4. SUFFICES: ??? by 10, $(\sigma, (\mathbf{fun}\, x : t \to e_1)\, v) \in \mathcal{C}_{j-1}[\![\theta(t')]\!]$.

⟨2⟩5. SUFFICES: ??? by 10, $(\sigma, e_1[x/v]) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$.

⟨2⟩6. By induction, we have $[\![\Theta; \Delta, g : t \multimap t'; x : t \vdash e : t']\!]$.

⟨2⟩7. Instantiate this with $\theta, j-2, \delta[g \mapsto \mathbf{fun}\, x : t \to e_1], \gamma = [x \mapsto v], \sigma$ (???).
PROVE: $(\sigma, \mathbf{fun}\, x : t \to e_1) \in \mathcal{V}_{j-2}[\![\theta(t) \multimap \theta(t')]\!]$.

⟨3⟩1. SUFFICES: ??? by 10, $(\sigma', e_1[x/v']) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$ for arbitrary $(\sigma', v') \in \mathcal{V}_{j-2}[\![\theta(t)]\!]$.

$\langle 3 \rangle 2$. We can again use the induction hypothesis $[\![\Theta; \Delta, g : t \multimap t'; x : t \vdash e : t']\!]$.

$\langle 3 \rangle 3$. But since it's true for $\mathcal{C}_0[\![\cdot]\!]$ (base case), it's true by induction ???

$\langle 2 \rangle 8$. Lastly, we show $\delta(\gamma(e)) = e_1[x/v]$, which follows by their definitions, to conclude $(\sigma, e_1[x/v]) \in \mathcal{C}_{j-2}[\![\theta(t')]\!]$.

$\langle 1 \rangle 13$. CASE: TY_VAR_LIN.
PROVE: $(\sigma, \gamma(\delta(x))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

$\langle 2 \rangle 1$. $\Gamma = \{x : t\}$ by assumption of TY_VAR_LIN.

$\langle 2 \rangle 2$. SUFFICES: $(\sigma, \gamma(x)) \in \mathcal{C}_k[\![\theta(t)]\!]$ by 3.

$\langle 2 \rangle 3$. By 2b, there exist $(\sigma_x, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]$, such that $\sigma = \sigma_x$ and $\gamma = [x \mapsto v_x]$.

$\langle 2 \rangle 4$. ??? Hence, $(\sigma_x, v_x) \in \mathcal{C}_k[\![\theta(t)]\!]$, by 6.

$\langle 1 \rangle 14$. CASE: TY_VAR.
PROVE: $(\sigma, \gamma(\delta(x))) \in \mathcal{C}_k[\![\theta(t)]\!]$.

$\langle 2 \rangle 1$. $x : t \in \Delta$ and $\Gamma = \emptyset$ by assumption of TY_VAR.

$\langle 2 \rangle 2$. SUFFICES: $(\emptyset, \delta(x)) \in \mathcal{C}_k[\![\theta(t)]\!]$ by 3 and 2b.

$\langle 2 \rangle 3$. By 2c, there exists v_x such that $(\emptyset, v_x) \in \mathcal{V}_k[\![\theta(t)]\!]$.

$\langle 2 \rangle 4$. ??? Hence, $(\emptyset, v_x) \in \mathcal{C}_k[\![\theta(t)]\!]$, by 6.

$\langle 1 \rangle 15$. CASE: TY_UNIT_INTRO.
PROVE: $(\sigma, \gamma(\delta(\,()\,))) \in \mathcal{C}_k[\![\theta(\mathbf{unit})]\!]$.

$\langle 1 \rangle 16$. CASE: TY_BOOL_TRUE, TY_BOOL_FALSE, TY_INT_INTRO, TY_ELT_INTRO.
Similar to TY_UNIT_INTRO.

5 Grammar Definition

m	$::=$			matrix expressions
	$\mid$	M		matrix variables
	$\mid$	$m + m'$		matrix addition
	$\mid$	$m\, m'$		matrix multiplication
	$\mid$	(m)	S	
f	$::=$			fractional capability
	$\mid$	fc		variable
	$\mid$	1		whole capability
	$\mid$	$\frac{1}{2} \cdot f$		
t	$::=$			linear type
	$\mid$	**unit**		unit
	$\mid$	**bool**		boolean (true/false)
	$\mid$	**int**		63-bit integers
	$\mid$	**elt**		array element
	$\mid$	f **arr**		arrays
	$\mid$	f **mat**		matrices
	$\mid$	$!t$		multiple-use type
	$\mid$	$\forall\, fc.t$	bind fc in t	frac. cap. generalisation
	$\mid$	$t \otimes t'$		pair
	$\mid$	$t \multimap t'$		linear function
	$\mid$	(t)	S	parentheses
p	$::=$			primitive
	$\mid$	**not**		boolean negation
	$\mid$	$(+)$		integer addition
	$\mid$	$(-)$		integer subtraction
	$\mid$	$(*)$		integer multiplication
	$\mid$	$(/)$		integer division
	$\mid$	$(=)$		integer equality
	$\mid$	$(\langle)$		integer less-than
	$\mid$	$(+.)$		element addition
	$\mid$	$(-.)$		element subtraction
	$\mid$	$(*.)$		element multiplication
	$\mid$	$(/.)$		element division
	$\mid$	$(= .)$		element equality
	$\mid$	$(< .)$		element less-than
	$\mid$	**set**		array index assignment
	$\mid$	**get**		array indexing
	$\mid$	**share**		share array
	$\mid$	**unshare**		unshare array
	$\mid$	**free**		free arrary
	$\mid$	**array**		Owl: make array
	$\mid$	**copy**		Owl: copy array
	$\mid$	**sin**		Owl: map sine over array

	|	**hypot**		Owl: $x_i := \sqrt{x_i^2 + y_i^2}$
	|	**asum**		BLAS: $\sum_i \lvert x_i \rvert$
	|	**axpy**		BLAS: $x := \alpha x + y$
	|	**dot**		BLAS: $x \cdot y$
	|	**rotmg**		BLAS: see its docs
	|	**scal**		BLAS: $x := \alpha x$
	|	**amax**		BLAS: argmax $i : x_i$
	|	**setM**		matrix index assignment
	|	**getM**		matrix indexing
	|	**shareM**		share matrix
	|	**unshareM**		unshare matrix
	|	**freeM**		free matrix
	|	**matrix**		Owl: make matrix
	|	**copyM**		Owl: copy matrix
	|	**copyM_to**		Owl: copy matrix onto another
	|	**sizeM**		dimension of matrix
	|	**trnsp**		transpose matrix
	|	**gemm**		BLAS: $C := \alpha A^{T?} B^{T?} + \beta C$
	|	**symm**		BLAS: $C := \alpha AB + \beta C$
	|	**posv**		BLAS: Cholesky decomp. and solve
	|	**potrs**		BLAS: solve with given Cholesky
v	$::=$			values
	|	p		primitives
	|	x		variable
	|	$()$		unit introduction
	|	**true**		true
	|	**false**		false
	|	k		integer
	|	l		heap location
	|	el		array element
	|	**Many** v		!-introduction
	|	**fun** $fc \to v$		frac. cap. abstraction
	|	$v[f]$		frac. cap. specialisation
	|	(v, v')		pair introduction
	|	**fun** $x : t \to e$	bind x in e	abstraction
	|	**fix** $(g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
	|	(v)	S	parentheses
e	$::=$			expression
	|	p		primitives
	|	x		variable
	|	**let** $x = e$ **in** e'	bind x in e'	let binding
	|	$()$		unit introduction
	|	**let** $() = e$ **in** e'		unit elimination
	|	**true**		true

	$\vert$	$\mathbf{false}$		false
	$\vert$	$\mathbf{if}\, e\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2$		if
	$\vert$	k		integer
	$\vert$	l		heap location
	$\vert$	el		array element
	$\vert$	$\mathbf{Many}\, e$		!-introduction
	$\vert$	$\mathbf{let}\,\mathbf{Many}\, x = e\, \mathbf{in}\, e'$		!-elimination
	$\vert$	$\mathbf{fun}\, fc \rightarrow e$		frac. cap. abstraction
	$\vert$	$e[f]$		frac. cap. specialisation
	$\vert$	(e, e')		pair introduction
	$\vert$	$\mathbf{let}\, (a, b) = e\, \mathbf{in}\, e'$	bind $a \cup b$ in e'	pair elimination
	$\vert$	$\mathbf{fun}\, x : t \rightarrow e$	bind x in e	abstraction
	$\vert$	$e\ e'$		application
	$\vert$	$\mathbf{fix}\, (g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
	$\vert$	(e)	S	parentheses
C	$::=$			evaluation contexts
	$\vert$	$\mathbf{let}\, x = [-]\, \mathbf{in}\, e$	bind x in e	let binding
	$\vert$	$\mathbf{let}\, () = [-]\, \mathbf{in}\, e$		unit elimination
	$\vert$	$\mathbf{if}\, [-]\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2$		if
	$\vert$	$\mathbf{Many}\, [-]$		!-introduction
	$\vert$	$\mathbf{let}\,\mathbf{Many}\, x = [-]\, \mathbf{in}\, e$		!-elimination
	$\vert$	$\mathbf{fun}\, fc \rightarrow [-]$		frac. cap. abstraction
	$\vert$	$[-][f]$		frac. cap. specialisation
	$\vert$	$([-], e)$		pair introduction
	$\vert$	$(v, [-])$		pair introduction
	$\vert$	$\mathbf{let}\, (a, b) = [-]\, \mathbf{in}\, e$	bind $a \cup b$ in e	pair elimination
	$\vert$	$[-]e$		application
	$\vert$	$v[-]$		application
Θ	$::=$			fractional capability environment
	$\vert$	$\cdot$		
	$\vert$	Θ, fc		
Γ	$::=$			linear types environment
	$\vert$	$\cdot$		
	$\vert$	$\Gamma, x : t$		
	$\vert$	Γ, Γ'		
Δ	$::=$			intuitionistic types environment
	$\vert$	$\cdot$		
	$\vert$	$\Delta, x : t$		
σ	$::=$			heap
	$\vert$	$\{\}$		empty heap
	$\vert$	$\sigma \uplus \{l \mapsto_f m_{k_1,k_2}\}$		location l points to matrix m

StepsTo	::=		result of small step
	$\mid$	$\langle \sigma, e \rangle$	heap and expression
	$\mid$	**err**	error